CLAIMS

What is claimed is:

1. A system to provide a network-accessible service, comprising:

an annotated source code, which is a programming language augmented with declarative meta-data capable of exposing program logic as a network-accessible service;

at least one deployed service component capable of providing the network-accessible service to a client; and

an enhanced compiler capable of analyzing the annotated source code, recognizing numerous types of meta-data annotations, and generating a mechanism, which can include one or more of: object files, software components and deployment descriptors, to facilitate the deployment of the at least one service component.

2. The system according to claim 1, wherein:

the network-accessible service is a Web service.

3. The system according to claim 1, wherein:

the system is capable of simultaneously managing multiple transactions, wherein each transaction can be a conversation of a request and/or a response from the client for the network-accessible service.

4. The system according to claim 3, wherein:

the system is capable of managing multiple asynchronous transactions, wherein within each asynchronous transaction, the response may be temporally separated from the initiating request for the network-accessible service from the client.

5. The system according to claim 1, further comprising:

an integrated development environment (IDE) capable of facilitating a graphical interface-based design and deployment of the network-accessible service.

6. The system according to claim 1, wherein:

the annotated source code is Java-based.

7. The system according to claim 1, wherein:

the meta-data can be either in-file with the annotated source code, or in a separate file, which can be a specially formatted XML file.

8. The system according to claim 1, wherein:

the annotated source code is capable of facilitating access to an external service, which can be one of stateful, stateless, synchronous, and asynchronous.

9. The system according to claim 1, wherein:

the annotated source code is capable of defining a wire binding between the network -accessible service and a physical wire format and/or protocol.

10. The system according to claim 9, wherein:

the wire binding can be at least one of:

SOAP over HTTP or SMTP;

transport of XML via generic HTTP Post;

transport of XML over other protocols such as FTP and mail; and

transport of XML over messaging services such as JMS or MSMQ.

11. The system according to claim 2, wherein:

the at least one service component comprises a servlet container and an Enterprise Java Bean (EJB) container, which are coupled together to deploy a Web service.

12. The system according to claim 11, wherein:

the servlet container is capable of at least one of:

listening and responding to a service request from the client; and

identifying and queuing the service request to be buffered.

13. The system according to claim 11, wherein:

the EJB container is capable of dispatching a service request based on meta-data to a stateless or stateful component.

14. The system according to claim 1, wherein:

the enhanced compiler is capable of creating, deploying and/or managing at least one security type using a specification provided by the annotated source code, wherein the at least one security type can be applied to one of: request message, callback, response message, and request message and/or callback originating with an intermediate service in a chain.

15. The system according to claim 14, wherein:

the at least one security type can be one of:

transport security for a data communications channel;

message based encryption security to protect headers and payloads in messages;

authentication of users or computing devices for channel or message based security;

digital signatures for verifying the originator of a message; and

user identity and declarative role-based security for granting or deny a client access to a particular service.

16. The system according to claim 1, wherein:

the enhanced compiler is capable of creating reliable messaging software for the network-accessible service using a specification provided by the annotated source code, wherein the reliable message software is capable of guaranteeing message delivery for communication between the service and the client.

17. The system according to claim 1, wherein:

the enhanced compiler is capable of creating an interceptor using a specification provided by the annotated source code, wherein the interceptor is capable of allowing at least one user-defined pre or post processing method to be applied to a message from the client.

18. The system according to claim 17, wherein:

the at least one processing method can be one of:

interception and transformation of message headers and contents;

message logging and auditing;

message redirection; and

protocol conversion.

19. A method to provide a network-accessible service, comprising:

exposing program logic as a network-accessible service using an annotated source code, which is a programming language augmented with declarative meta-data;

providing the network-accessible service to a client via at least one deployed service component; and

analyzing the annotated source code, recognizing numerous types of meta-data annotations, and generating a mechanism, which can include one or more of: object files, software components and deployment descriptors, to facilitate the deployment of the at least one service component.

20. The method according to claim 19, wherein:

the network-accessible service is a Web service.

21. The method according to claim 19, further comprising:

simultaneously managing multiple transactions, wherein each transaction is a conversation of a request and/or a response from the client for the network-accessible service.

22. The method according to claim 21, further comprising:

managing multiple asynchronous transactions, wherein within each asynchronous transaction, the response may be temporally separated from the initiating request for the network-accessible service from the client.

23. The method according to claim 19, further comprising:

facilitating a graphical interface-based design and deployment of the network-accessible service.

24. The method according to claim 19, further comprising:

facilitating access to an external service, which can be one of stateful, stateless, synchronous, and asynchronous.

25. The method according to claim 19, further comprising:

defining a wire binding between the network-accessible service and a physical wire format and/or protocol.

26. The method according to claim 20, further comprising:

listening and responding to a Web service request from the client; and

identifying and queuing the Web service request to be buffered.

27. The method according to claim 20, further comprising:

dispatching a Web service request based on meta-data to a stateless or stateful component.

28. The method according to claim 19, further comprising:

creating, deploying and/or managing at least one security type using a specification provided by the annotated source code, wherein the at least one security type can be applied to one of: request message, callback, response message, and request message and/or callback originating with an intermediate service in a chain.

29. The method according to claim 19, further comprising:

creating reliable messaging software for the network-accessible service using a specification provided by the annotated source code, wherein the reliable message software is capable of guaranteeing message delivery for communication between the service and the client.

30. The method according to claim 19, further comprising:

creating an interceptor using a specification provided by the annotated source code, wherein the interceptor is capable of allowing at least one user-defined pre and post processing method to be applied to a message from the client.

31. A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

    expose program logic as a network-accessible service using an annotated source code, which is a programming language augmented with declarative meta-data;

    provide the network-accessible service to a client via at least one deployed service component; and

    analyze the annotated source code, recognize numerous types of meta-data annotations, and generate a mechanism, which can include one or more of: object files, software components and deployment descriptors, to facilitate the deployment of the at least one service component.

32. The machine readable medium of claim 31, wherein:

    the network-accessible service is a Web service.

33. The machine readable medium of claim 31, further comprising instructions that when executed cause the system to:

    simultaneously manage multiple transactions, wherein each transaction is a conversation of a request and/or a response from the client for the network-accessible service.

34. The machine readable medium of claim 32, further comprising instructions that when executed cause the system to:

    manage multiple asynchronous transactions, wherein within each asynchronous transaction, the response may be temporally separated from the initiating request for the network-accessible service from the client.

35. The machine readable medium of claim 31, further comprising instructions that when executed cause the system to:

    facilitate a graphical interface-based design and deployment of the network-accessible service.

36. The machine readable medium of claim 31, further comprising instructions that when executed cause the system to:

facilitate access to an external service, which can be one of stateful, stateless, synchronous, and asynchronous.

37. The machine readable medium of claim 31, further comprising instructions that when executed cause the system to:

define a wire binding between the network-accessible service and a physical wire format and/or protocol.

38. The machine readable medium of claim 32, further comprising instructions that when executed cause the system to:

listen and respond to a Web service request from the client; and
identify and queue the Web service request to be buffered.

39. The machine readable medium of claim 32, further comprising instructions that when executed cause the system to:

dispatch a Web service request based on meta-data to a stateless or stateful component.

40. The machine readable medium of claim 31, further comprising instructions that when executed cause the system to:

create, deploy and/or manage at least one security type using a specification provided by the annotated source code, wherein the at least one security type can be applied to one of: request message, callback, response message, and request message and/or callback originating with an intermediate service in a chain.

41. The machine readable medium of claim 31, further comprising instructions that when executed cause the system to:

create reliable messaging software for the network-accessible service using a specification provided by the annotated source code, wherein the reliable message software is capable of guaranteeing message delivery for communication between the service and the client.

42. The machine readable medium of claim 31, further comprising instructions that when executed cause the system to:

create an interceptor using a specification provided by the annotated source code, wherein the interceptor is capable of allowing at least one user-defined pre and post processing method to be applied to a message from the client.

43. A system to provide a network-accessible service, comprising:

means for exposing program logic as a network-accessible service using an annotated source code, which is a programming language augmented with declarative meta-data;

means for providing the network-accessible service to a client via at least one deployed service component; and

means for analyzing the annotated source code, recognizing numerous types of meta-data annotations, and generating a mechanism, which can include one or more of: object files, software components and deployment descriptors, to facilitate the deployment of the at least one service component.

44. A computer data signal embodied in a transmission medium, comprising:

a code segment including instructions to expose program logic as a network-accessible service using an annotated source code, which is a programming language augmented with declarative meta-data;

a code segment including instructions to provide the network-accessible service to a client via at least one deployed service component; and

a code segment including instructions to analyze the annotated source code, recognize numerous types of meta-data annotations, and generate a mechanism, which can include one or more of: object files, software components and deployment descriptors, to facilitate the deployment of the at least one service component.